# Parameters and Hyperparameters

1. Params in ML models

* The objective of a typical learning algorithm is to find a function f that **minimizes a certain loss** over a dataset
* The learning algorithm produces f through the **optimization of a training criterion** with respect to a set of params

1. Linear regression params

* **β**, the coefficients of the linear function, are the **parameters** to find or optimize by the algorithm

1. Decision tree parameters

* The variable
* The split value
* The height in the tree

1. Neural network parameters

* The weights at each neuron

1. Hyperparameters in the ML models

* Hyperparams are params that are **not directly learned** by the learning algorithm
* Hyperparams are **specified outside** of the training procedure
* Hyperparams control the **capacity** of the model, i.e., how flexible the model is to fit the data
* Prevent over-fitting

1. Linear regression hyperparameters

* Vanilla Linear regression: No hyperparameters
* Regularized linear regression:
* The regularization method:
  + Lasso
  + Ridge
  + Elastic net
* The regularization penalty

1. Decision tree hyperparameters

* The metric to measure the quality of the split
* The number of features to evaluate at each node
* The depth of the tree
* The minimum number of samples required to split the data further
* Etc.

1. Random forests and GBMs

* Number of trees (or estimators)
* Learning rate (GBMs)

1. Neural network hyperparams

* Number of layers
* Number of neurons per layer
* The activation function
* Dropout rate
* Etc.

1. Other model hyperparams

* Nearest neighbors -> number of neighbors
* SVMs -> the kernel function

1. Hyperparameters in ML models

* Hyperparams could have a big impact on the performance of the learning algorithm
* Optimal hyperparams settings often differ for different datasets
* Therefore they should be optimized for each dataset

# Hyperparameter Optimization

1. Parameters vs. Hyperparameters

|  |  |
| --- | --- |
| **Parameters** | **Hyperparameters** |
| Intrinsic to model equation  Optimized during training | Defined before training  Constrain the algorithm |

1. Random forests and GBMs – Hyperparams (see previous section)
2. Effect of hyperparameters

* Fit several GBMs with different hyperparams
* Measure each model performance -> rmse
* Certain hyperparams return models with decreased performance
* More than 1 combination of hyperparams return a good fit

1. Hyperparameter optimization

* The process of finding the best hyperparams for a given dataset is called Hyperparameter Optimization or Hyperparameter Tuning
* Methods to choose the hyperparams that minimize the generalization error (not necessarily the loss)

1. Challenges

* We can’t define a formula to find the hyperparms
* Try different combinations of hyperparams and evaluate model performance
* The critical step is to choose how many different hyperparam combinations we are going to test
* Increased number of hyperparam combinations -> increased chance to get a better model -> increased computational cost

1. Methods

* How do we find the hyperparam combinations to maximize performance while diminishing computational costs
* Different hyperparam optimization strategies
* Methods
* Manual search
* Grid search
* Random search
* Bayesian optimization
* Etc.

1. Search

* A search consists of:
* Hyperparam space
* A method for sampling candidate hyperparams
* A cross-validation scheme
* A performance metric to minimize (or maximize)

1. Hyperparam response surface

* Find the hyperparams that minimize (or maximize) a performance metric
* Hyperparams = min(performance metric)
* Response surface - Ψ(λ):
* Algorithm
* Hyperparameters
* Dataset
* Metric

1. Low effective dimension

* Ψ(λ) are more sensitive to changes in some dimensions
* Most params do not matter much